5-1-1999

Hieroglyphs for the information age: Images as a replacement for characters for languages not written in the Latin-1 alphabet

Akira Hasegawa

Hieroglyphs for the Information Age: Images as a Replacement for Characters for Languages not Written in the Latin-1 Alphabet

by

Akira Hasegawa

A thesis project submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Printing Management and Sciences in the College of Imaging Arts and Sciences of the Rochester Institute of Technology

May, 1999

Thesis Advisor: Professor Frank Romano

School of Printing Management and Sciences
Rochester Institute of Technology
Rochester, New York

Certificate of Approval

Master's Thesis

This is to certify that the Master's Thesis of

Akira Hasegawa

With a major in Graphic Arts Publishing
has been approved by the Thesis Committee as satisfactory
for the thesis requirement for the Master of Science degree
at the convocation of

May 1999

Thesis Committee:

Frank Romano
Thesis Advisor

Marie Freckleton
Graduate Program Coordinator

C. Harold Goffin
Director or Designate

Hieroglyphs for the Information Age:

Images as a Replacement for Characters for Languages not Written in the Latin-1 Alphabet

I, Akira Hasegawa, hereby **grant permission** to the Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

May 20, 1999
Date

Signature of Author

Acknowledgements

First of all, I would like to express my appreciation to my thesis adviser, Prof. Frank Romano, for his suggestion of this topic and guidance throughout this thesis study. Appreciation is extended to Prof. Marie Freckleton for assisting in the academic affairs. Finally, I would like to thank my family and friends for their encouragement. Without them, completion of this thesis would be impossible.

Table of Contents

List of Tables

Abstract

One of the biggest existing problems for international users and international companies is the difficulty of displaying web pages written in multiple languages. This thesis project demonstrates a solution in which the problems of text encoding are eliminated by representing all of the text on a page as a series of image files. The Japanese text on several Japanese web pages was converted into glyphs, which were defined as image files containing images of text. The glyphs were arranged on new web pages, so that the same Japanese text can be read by any web surfer, regardless of the operating system or fonts available on his or her computer. Statistics are given to demonstrate that the proposed method can be used to generate glyph-based web pages quickly, that glyphs can be downloaded more quickly than other types of graphics (e.g. photographs and computer animation), that users with Japanese-language software on their computers find no significant difference between the text-based and glyph-based web pages, and that users without Japanese-language software were able to view the glyph-based web pages in Japanese. The new web pages are available for public download, at http://www.kcg.edu/event2 (GIF glyphs) and http://www.kcg.edu/event3 (JPEG glyphs).

Chapter 1

Introduction

One of the biggest existing problems for international users and international companies is the difficulty of displaying Web pages written in multiple languages. If a user wants to connect to a Japanese language site, the user has to have a Japanese Web browser, with Japanese fonts installed on the computer; on most computers, the user needs, in addition, an operating system localized to the Japanese language. Likewise, if a user wants to display Cyrillic or Greek or Arabic characters, appropriate software and fonts are necessary. For most users, the time and expense required to buy or download new fonts and system extensions for each new language is prohibitive. Thus most users are only able to view Web pages written in the character set pre-loaded on the computer; Web pages in other languages appear as meaningless strings of Roman characters.

This thesis demonstrates a solution in which the problems of text encoding are eliminated by representing all of the text on a page as a series of image files. For convenience, image files containing images of text will be referred to in this thesis as hieroglyphs, or glyphs for short. This thesis project describes experiments in which the Japanese text on several Japanese Web pages were converted into glyphs, and arranged on new Web pages, so that the same Japanese text can be read by any Web surfer, regardless of the operating system or fonts available on his or her computer.

This thesis project describes a project in which all of the Japanese-language text in an existing tree of Web pages was converted into image files called glyphs, so that it can be displayed on computers without installed Japanese language software.

The text on a set of existing Japanese-language Web pages was displayed separately from any background and embedded images, and captured as images in both the CompuServe Graphical Interchange Format (GIF) and the Joint Photographic Experts Group format (JPEG)[1]. The captured images were laid out on two new sets of Web pages (one set using GIF images, one set using JPEG images), overlaid on top of the original background images, and appropriately interspersed with the original foreground images.

In order that the reader may better understand the motivation and significance of this work, chapter two will review the history of some of the relevant technologies, including the internet, the American Standard Code for Information Interchange (ASCII) and unicode text representation standards, and the GIF and JPEG image representation standards. Chapter three reviews the objectives of this thesis. Chapter four describes the methods used to capture text as images, and to format new Web pages using the captured images; special attention is given to the time required to convert a text-based Web page into a glyph-based Web page. Chapter five describes the results of both objective experiments and subjective surveys designed to measure glyph download time, ease of use, and general user acceptance. Chapter six summarizes the conclusions, and proposes directions for future research

Endnotes for Chapter 1

[1] Greenspun, Philip. Philip and Alex's Guide to Web Publishing. New York, NY: Morgan Kaufman, 1999.

Chapter 2

Review of Literature in the Field

History of the Internet

Nowadays, the Internet has become well known. People think it is new technology, but it has been around for while. It was only used for researchers, military and government personnel.

The ARPANET

At the beginning of the cold war, during the 1960's the United States Government was faced with a problem: How could the United States authorities communicate with each other in the aftermath of a nuclear attack?

Networks were linked directly at each location point-to-point. This means that if one point in the network was destroyed, the whole network would become useless. To solve this problem, the Department of Defense's Advanced Research Projects Agency (ARPA) created the network called ARPANET that was a non-centralized network. This was not like the original network which was based on a point-to-point network. Therefore the new network would work even if one point of a network was destroyed. Another advantage was that there are no main centers in the ARPANET network, there would be no primary target for enemies to destroy. The ARPANET was also intended to encourage researchers in the United States to share super computers with each other.

In 1969, researchers at four universities in the United States become the first hosts of the ARPANET. They were Stanford Research Institute, University of California at Los Angeles, University of California at Santa Barbara and the University of Utah.[1]

During the 1970's, more universities and government research centers around the United States were connected as hosts to the ARPANET. The ARPANET was originally designed for researchers and government people to share data and access remote computers, but the use of electronic mail applications caused ARPANET to become a high-speed electronic post office. Mailing lists were developed and people were able to send and receive massages twenty-four hours a day via electronic mail. People started using the ARPANET not only to collaborate on research notes but also to discuss various topics of interest.

By the middle of the 1970's, the commercial version of the ARPANET went on line and the ARPANET started to move away from its own military and research roots towards use by the general public. As ARPANET become larger and more sophisticated, a standard protocol was needed. In 1977, the protocol was invented and called TCP/IP. This protocol lets the user in a small network connect to the main network, ARPANET.

TCP/IP and the Internet boom

The TCP/IP (Transmission Control Protocol and Internet Protocol) protocol can be used by any kind of computer. For this reason, it was easy for other computer networks (university networks, corporate networks, etc.) to talk to ARPANET, and thus to each other through ARPANET, by using TCP/IP. By the late 1970s, there were several out-

side networks linked to ARPANET. By the early 1980s, this "network of networks" was being called the "Internet" (inter = between), because it was the network which sat between other networks, keeping them connected. In the early 1990's, several companies (including BBN Corporation, UUNet Technologies, America On Line, and most of the world's telephone companies, but especially Mass Communication Incorporated) began building their own fiber optic "backbone" networks, and charging for network access. By 1995, these networks were so large that NSFNET stopped carrying Internet traffic, and was converted back into a research project.[2]

In 1981, there were 213 hosts to ARPANET. Approximately once every twenty days a new host was added. For the first time the loose collection of networks using TCP/IP which made up the ARPANET was seen as an "Internet".[3]

In 1982, the term "Internet" was used for the first time. In 1983, the TCP/IP was made a standard.

By the middle of the 1980's, the personal computer and super-minicomputer industries had grown rapidly, making less expensive, smaller and more powerful computers each year. Some computers came with network ready features. Because of these computers, many companies had a chance to use the Internet for the first time. Some corporations started to use the Internet as a tool to communicate with each other and with their customers.

The National Science Foundation NET (NSFNET) was started in 1985 by the United States National Science Foundation. Basically, the NSFNET was designed to connect

the same supercomputers and universities as ARPANET, but it was supposed to connect them much faster, using optical fiber cables called T1 cables which carried data at 1.5 Mbps (million bits per second). This network of high-speed optical cables was the first "backbone network." (Nowadays, in 1996, any business can get a T1 Internet connection for about 2500 dollars/month, while the "backbone networks" run at 50 Mbps.[4]) The NSFNET was so successful that in 1989, ARPANET disbanded, and most of the equipment and cables from their system were given to NSFNET.[5]

Internet Software

Finally, there is the question of which software uses the Internet. In the beginning, the Internet was used mostly for e-mail, network news, remote computing (using Telnet and other programs), and file transfers. The file transfer protocol (ftp) gave rise to a new use for the Internet. Big databases were made available on the Internet, and anyone who wanted files from the database could log onto the net using the name "anonymous" and collect whatever files they wanted. Unfortunately, it was difficult to find the files you wanted. The first program which made Internet data available using a "point-and-click" interface was "Gopher," developed at the University of Minnesota.

The application which really caught on, however, was the World Wide Web. The World Wide Web is not a program, or a communication protocol, as much as it is a metaphor. All of the information on the Internet is presented to the user as a single, gigantic hypermedia "help file." The first Web programs, designed to read an extremely simple text formatting language called HyperText Markup Language (HTML), were posted to the group alt.hypertext by CERN's (European Laboratory for Particle Physics) Tim Berners-Lee in 1991. Students and staff at the National Center for

Supercomputer Applications located on the campus of University of Illinois at Urbana Champaign picked up the idea, and rapidly developed a hypertext server program (HTTPD - HyperText Transfer Protocol Daemon) and a fully graphical hypertext browser program (Mosaic), both of which were made available free of charge, pre-compiled for most common computer architectures. These plug-and-play applications made it easy for people to publish their own Web pages, complete with pictures and sound. Since publishing on the Web was easy, many people and companies did it, and since so much information was suddenly available on the Web, many consumers decided to get an Internet connection. The person who wrote Mosaic at University of Illinois was a student named Marc Andreeson, who then graduated and went on to found Netscape. In 1995, the Internet programming language called Java was released by Sun Microsystems. Java changed the way information and applications could be retrieved, displayed, and used over the Internet. Between 1994 and 1996, more than one billion dollars per year exchanged hands at Internet shopping malls, and the number of computers connected to the Internet grew several hundred percent per year.[1,4]

The number of Internet users is estimated to double every year. By the year 2020, it is estimated that everyone would have e-mail address. The main cause of this Internet user increase was the invention of the World Wide Web.[6]

Representation of Text Information

8-bit Character Sets

ASCII stands for "American Standard Code for Information Interchange." ASCII was adopted as the standard code for non-numerical information by a large number of early computer manufacturers, and thus became a de facto standard. The leading com-

petitor to ASCII was a system called EBCDIC (extended binary coded decimal) developed and promoted by IBM; since most computers built by other companies could only communicate in ASCII, IBM was eventually forced to abandon EBCDIC.

The basic ASCII set is 128 codes (7 bits of information). The reason for this is that in old network protocols, one bit out of every eight had to be zero, in order to keep the communication hardware synchronized.

In 1987, the International Standards Organization began issuing a set of standard 8-bit, 256-character alphabets for the digital communication of information.[7] The first character set, called "Latin alphabet number 1" (standard 8859-1), was designed as an extension of the ASCII character set, with an additional 128 characters defined to allow digital representations of non-English languages written in the Latin alphabet, such as the German "double s." The Latin 1 alphabet is probably, at present, the most widely used text encoding system in the world.

The International Standard Organization (ISO) standard "Latin alphabet number 2" (standard 8859-2) is an extended character set designed to encode slavic languages written in a Latin alphabet, including Czech, Slovak, Polish and Hungarian. The Latin/Cyrillic alphabet (standard 8859-5) is a 256-character code containing all of the standard characters used in both the Latin and Cyrillic alphabets. Similar mixed alphabets include the Latin/Arabic alphabet (standard 8859-6), the Latin/Greek alphabet (standard 8859-7), the Latin/Hebrew alphabet (standard 8859-8), and the Latin/Celtic alphabet (standard 8859-14).

In all, the ISO issued fifteen standard 8-bit alphabets between 1987 and 1992. Each of these alphabets represents the characters in a small subset of the world's languages using at most 256 characters. This system is obviously inefficient for documents intended for an international audience; in order to include, say, Greek and Cyrillic text in a single document, it is necessary to switch somehow between two different ISO coding schemes. Languages with more than 256 characters, including Chinese, Japanese, and Korean, can't be encoded using any 8-bit coding system. For these reasons, the ISO and the computer manufacturing community began considering double-byte and multi-byte representations of text.

Japanese Character Sets

Standard written Japanese uses 142 phonetic characters (if diacritically-marked characters are counted separately, e.g. if "ba" is counted separately from "pa"), and approximately 2000 semantic characters. Since it is impossible to represent Japanese using an 8-bit code, standard text-encoding schemes in Japan use codes which use either a fixed 16-bit word per character (the original Japan Industrial Standards (JIS)) or a mixture of 8-bit and 16-bit codes (Shift-JIS and Extended Unix Code (EUC)).

Japanese text encoding schemes are generally based on the two-byte standard codes established by the Japan Standards Association. There are three revisions of the standard: JIS 0208-1978, JIS 0208-1983, and JIS 0208-1990.[8] The JIS standards encode every character using two bytes, but like ASCII, JIS is a seven-bit standard: the top bit of every byte is always assumed to be zero. For this reason, JIS might be called a 14-bit character set.

Documents written in Japanese often include both Japanese and Latin characters, and programming languages usually consist of primarily ASCII characters, with Japanese characters only in the comments. Programs containing many Latin characters may be stored more efficiently using a multibyte representation, in which ASCII characters are stored using one byte, and Japanese characters are stored using two bytes. There are several multibyte character sets available, depending on the desired application.

Internal representations on both unix and Microsoft operating systems use encoding systems which contain the entire ASCII character set as a subset. For example, Microsoft computers use the "shift-JIS" representation, in which the length of a character is specified by the first bit of the character: if the first bit is zero, the character is part of the normal ASCII character set, while if the first bit is one, the character is a 16-bit representation of a Japanese character. For example, the first bit of 0x1C is zero, so 0x1C is an ASCII character (in this case the character '\'), while the first bit of 0x9C is one, so 0x9C must be the first half of a two-byte Japanese character.[9]

The problem with the shift-JIS system is that the second byte of a Japanese character may be anything; for example, the two-byte sequence 0x9C1C is a legal shift-JIS encoding of a Japanese character. Unfortunately, older software written in the United States may not even consider the possibility that characters might be two bytes in length. An older program written in the United States might mis-interpret the code 0x9C1C as a sequence of two characters: it would not know what the first character is, but it would "know" that the second character is '\'. On a DOS system, '\' is used to separate directories in a path name, so the older DOS program might mistakenly interpret everything after the 0x1C as a subdirectory.

Unix computers avoid this problem by specifying a code called EUC-JP (Extended Unix Code for Japan) in which the top bit is set in both bytes of a two-byte character.[10] Older programs written for ASCII will never mis-interpret the second byte of a two-byte EUC character: since both bytes have the top bit set, neither the first nor the second byte is a legal ASCII character by itself. In fact, the EUC code for any Japanese character is the same as the JIS code, but with the top bit set in both bytes: thus the EUC code 0x9C9C would represent the same character as the JIS code 0x1C1C.

EUC and shift-JIS codes are useful for internal encodings. For e-mail, however, they have a problem: many older e-mail programs are designed for the 7-bit ASCII representation, so they may automatically set the top bit of every byte to zero. For this reason, it was felt that e-mail applications needed a mixed encoding of Japanese and ASCII characters in which the top bit of every byte could be zero. The ISO-2202-JP standard is a code shifting standard established by Request for Comment (RFC) 1468[11] which has been accepted by the Internet Engineering Task Force as a Multipurpose Internet Mail Extensions (MIME) encoding standard for multimedia Internet documents. In ISO-2202-JP encoded documents, the document always starts out with text encoded in ASCII. The encoding is then switched to one of the 14-bit JIS standards using a special "mode-shifting character." The mode must be switched back to ASCII (or to a one-byte subset of JIS) at the end of every line of text; then the mode can be switched to two-byte codes again at the beginning of the next line.

It should be clear, from this discussion, that localizing English-language programs or software to the Japanese language is not an easy task. There are documents on the

world wide Web coded in JIS, shift-JIS, and EUC text codes, and the Japanese text embedded in these documents may be encoded using the 1978, 1983, or 1990 version of the JIS 0208 standard. In order to correctly display Japanese documents, a Web browser must be able to correctly decode all of these standard encoding systems.

Unicode

Unicode is an international character-encoding system designed to replace ASCII.[12,13] It is a fixed-width, sixteen-bit code that represents more than 65,000 characters.[14] Unicode represents nearly every language in modern use, as well as mathematical and technical symbols, and a few ancient languages.

"The Unicode Consortium, a nonprofit organization in Mountain View, California, was founded in 1991 to develop and promote the use of the Unicode standard. Charter members include Apple, Xerox, IBM, Microsoft, Sun Microsystems, and Novell. The ISO, based in Geneva, Switzerland, approved Unicode in June 1992 as the international character-encoding standard".[15]

The Unicode standard has significant advantages. The code makes it easy to specify modified characters or special cases. The code also has special control characters to handle changes in text direction within a single line of text.

There is a minor penalty for storing uniform two byte values for each character, however it eliminates the confusion of overlapping-single byte code pages in which a character's identity is dependent on the active code page. In contrast to multibyte character encodings, the uniform sixteen bit code of each character makes it easier to identify character boundaries.

The amount of new software using Unicode is promising. However, converting existing software to Unicode is arduous and in many cases not feasible. Additionally, because Unicode's design requires that every character be two bytes long, most programs will become larger. The converted program may also need additional memory just to operate. Therefore it may take sometime before the Unicode standard becomes widely accepted and implemented.

One of largest setbacks to Unicode being accepted is that many operating platforms do not support it. Presently only Windows NT and IBM's AIX support the code. Microsoft, by not incorporating Unicode in Window 95, caused a major setback to the acceptance of Unicode. Nevertheless, several major platforms plan on incorporating full support for Unicode sometime in the future. Developers can still write Unicode-based software for a non-Unicode platform, but it would be much easier if the operating system supported the Unicode standard.

Unicode has been incorporated into Sun's Web programming language, Java. It will also soon be incorporated into HTML (HyperText Markup Language). However only when Unicode becomes an industry-wide standard will the global-ready product design ever be achieved. There is some slow progress in this area. According to Freytag, "The first round of office-type application with Unicode support will be available in less than two years".[16] The international significance of having unicode available for users in two years time would be substantial. For example, multinational corporations would be able to transfer files from different applications without having to maintain code-page dependencies.

Summary

The 256-character Letin 1 alphabet (extended ASCII) is most likely to be in use for a considerable period of time. Windows 95, the current Macintosh operating system and most unix operating system will not include native support for the international Unicode standard in the near future. The success of the World Wide Web has been the primary force behind the incorporation of Unicode into Java, but HTML does not yet support Unicode. As corporate strategies for operating system development releases become more cautious and are primarily aimed at the United States users' convenience, it is not unreasonable to suppose that widespread implementation of Unicode is several years away from broad based acceptance.

The problem of displaying non-Roman alphabet characters on Latin-1 machines without specialized sub-operating systems or language kits will be unsolved for the near future.

Image Encoding Systems

Images can be encoded as bitmaps or pixel-maps of different resolutions, or they can be compressed in some way. Different applications and different types of images require different types of compression, so there are several image file formats in common use on the Internet. Most Web browsers support the GIF (Graphical Interchange Format, a trademark of CompuServe) and JPEG (Joint Photographic Experts Group) formats, as well as one or more bitmap formats which may depend on the operating system of the computer, so an image-based Web page should generally be presented in either GIF or JPEG format.

There are several types of pixel-map image formats available, including the Macintosh PIC format, the Microsoft BMP format, the X11 Consortium's XBM format, and the unix portable pixelmap format.[17] All pixel-mapped images are represented as a series of points, called pixels (picture elements), which are usually ordered in a rectangular grid scanned from left to right, and from top to bottom (although the scanning order varies depending on the file format). Each pixel is represented by either one integer (for a grayscale image) or a vector of three integers (for a color image). The maximum possible size of the integer pixel values determines the color depth. Typical color depths used on the Internet vary between 1 bit (for a black-and-white "bitmapped" image, as in the X11 Bitmap format) and 24 bits.

The CompuServe GIF format compresses image files using a type of run-length coding. Regions of similar color in an image are set to be exactly the same color, and the image file stores information about the color and boundaries of each such homogenous region. The GIF file format can represent an image with at most 256 distinct colors. The GIF format is useful for pictures with few colors, including line drawings.[18]

The JPEG format is an open standard for the compression of digitized photographs, proposed by the Joint Photographic Experts Group.[19] The JPEG format compresses a pixel map by smoothing off the edges inside small 256-pixel blocks. Files compressed using JPEG generally look better than files compressed using GIF if the original image is a smoothly varying scene, such as a landscape photograph, but JPEG generally looks worse than GIF if the original has many sharp edges.

An alternative file format which is not yet implemented in many Web browsers, but which may be soon, is the Portable Network Graphics (PNG) standardized format proposed by the World Wide Web Consortium.[20] PNG was proposed as a non-patented alternative to GIF, and has many of the same features, but PNG can represent more colors than GIF. PNG also allows explicit specification of the photographic gamma, so that it is possible for a display program to reproduce images with the same color intensity as the original; independent specification of gamma is not possible in either the JPEG or GIF formats.

Endnotes for Chapter 2

[1] Kalakota, Ravi. and Whinston, Andrew B. Electronic Commerce. Essex, England: Addison Wesley Longmon, 1997.

[2] Washburn, Kevin. and Evans, Jim. TCP/IP: Running a Successful Network. Essex, England: Addison Wesley Longmon, 1996.

[3] Randall, Nail. The Soul of the Internet. London, England: International Thomson Computer Press, 1997.

[4] Ellsworth, Jill H. , and Ellsworth, Matthew V. Internet Business Book, New York, NY: John Wiley and Sons, Inc., 1996.

[5] Benett, Gordon. Introducing Internets. Hollis, New Hampshire: Que Corporation, 1996

[6] Hahn. Harley. The Internet Complete Reference. Berkeley, California: Osborne McGraw-Hill, 1996.

[7] International Standards Organization, "Information processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1." ISO Standard 8859-1, 1987.

[8] Japanese Standards Association, "Code of the Japanese graphic character set for information interchange," JIS X 0208-1978, -1983 and -1990.

[9] Fowles, Ken. "Developing TrueType." Redmond, Washington: Microsoft Corporation, June 1997. http://www.microsoft.com/truetype/unicode/cs.htm

[10] Turnbull, Stephen. "Alphabet Soup: The Internationalization of Linux, Part 1." Linux Journal. Seattle, WA: Specialized Systems Consultants, Inc., March 1999.

[11] Murai, Jun., Crispin, Mark. and van der Poel, Erik M., "Japanese Character Encoding for Internet Messages," RFC 1468. Los Angeles: RFC Editor, Information Sciences Institute, June 1993. http://www.faqs.org/rfcs/rfc1468.html.

[12] International Standards Organization, "Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane." ISO Standard 10646-1, 1993

[13] Unicode Consortium. Unicode. San Jose, California: Unicode, Inc., 1992-1998. http://www.unicode.org/index.html

[14] Abramson, Dean, "Globalization of Windows," Byte Magazine. New York, NY: McGraw-Hill, November 1994. http://www.byte.com/art/9411/sec9/art7.htm

[15] Miller, L. Chris, "Transborder Tips and Traps," Byte Magazine. New York, NY: McGraw-Hill, June 1994. http://www.byte.com/art/9406/sec7/art2.htm

[16] Hars, Adele. "Organizing Babylon," Byte Magazine. New York, NY: McGraw-Hill, March 1996. http://www.byte.com/art/9603/sec18/art1.htm

[17] Poskanzer, Jeff. "ppmtopgm manual page." December, 1988.

[18] Greenspun, Philip. Philip and Alex's Guide to Web Publishing. New York, NY: Morgan-Kaufman, 1999.

[19] Lane, Thomas. ed. "JPEG image compression FAQ." Independent JPEG Group, October, 1997. http://www.cis.ohio-state.edu/hypertext/faq/usenet/jpeg-faq/index.html

[20] Boutell, Thomas ed., "PNG (Portable Network Graphics) Specification." Cambridge, MA: World Wide Web Consortium, October, 1996. http://www.boutell.com/boutell/png/index.html

Chapter 3

Statement of the Problem

These past few years the Internet has become one of the most rapidly growing of information media. The Internet needs continued improvement to meet all the needs of all users.

The Internet has changed people's life styles. By using the Internet, people can research or even shop from home using a computer. The Internet user can go anyplace in cyberspace to research or to get information. But if a user on a Latin-1 computer (typically, any computer sold in the United States or Western Europe) goes to a non-Roman alphabet language site, the user cannot understand the information, because the computer is not capable of displaying non-Roman characters. Instead of correctly displaying the non-Roman characters, the computer simply interprets and displays each byte on the page as the corresponding character in the Latin-1 character set, thus converting meaningful non-Roman text into a meaningless stream of Roman characters, as shown on the Web page in Appendix A. Thus, even if the user understands multiple languages, he or she is unable to read Web pages in those languages, because of limitations in the software.

In order to view multiple languages on a single computer, the user needs to install multiple language kits or multiple "localized" versions of the operating system. Using

multiple copies of the operating system requires a huge amount of disk space; typically, each version of the operating system must be stored on a separate hard disk partition, so that the available space on the hard disk is quickly used up. In addition, if the user wishes to switch from one version of the operating system to another, he must reboot the computer. Naturally, it is usually not practical to reboot the computer just because you want to look for a piece of information on a Japanese Web page.

Unfortunately, installing multiple language kits under a single operating system is also often not practical. First, many operating systems do not even support multiple language kits. Those operating systems which support multiple language kits often run more slowly and less reliably with multiple language kits installed.

Finally, for most users, the time and expense required to purchase, download, and install multiple language kits is prohibitive. In order to correctly view Web pages in all the languages of the world, at least twenty different language kits would be necessary (there are fifteen different ISO-standard 8-bit character codes, and none of the 8-bit codes covers Japanese, Chinese, Korean, or Thai). It is difficult to imagine a user willing to spend money and time to buy twenty different language kits; most of the world's computers, therefore, do not correctly display foreign-language Web pages.

Computers sold in countries which do not use the Latin-1 character set will usually display both the native character set and the Latin-1 character set pretty reliably, but there are still problems. First, getting such a computer to display some other character set (for example, getting a Japanese computer to display an Arabic character set) is at least as difficult as getting a Latin-1 computer to display the same character set.

Second, non-Latin-1 versions of most operating systems come out months or years after the Latin-1 versions, and are more expensive than the Latin-1 versions. Third, many application programs are never ported to non-Latin-1 character sets, and using an unported application on a non-Latin-1 computer can result in bizarre and buggy behavior.

Finally, there are some software programs that translate other languages into English. If these translation programs worked reliably, it could be argued that we would have no need to display multiple languages on a single computer; if a user wants information written in Japanese, he would be able to have it translated into something that both he and his computer could read. Unfortunately, automatic translation is still pretty buggy; even the manufacturers only advertise a "correct translation" rate of about 70-80 percent.

The goal of this thesis project is to test the use of images as replacements for ASCII characters for languages not written using the Latin-1 character set. As a test case, a Japanese-language Internet site was developed which can be viewed, in Japanese, by users who do not have a Japanese operating system (or Japanese language kit) on their computers. Chapters four and five of this thesis describe experiments designed to show that:

1. Any computer system using a Latin alphabet can be made to display non-Latin alphabets through the use of glyph image files.

2. Multilingual Web surfers (in this case, speakers of both Japanese and English) can browse Web sites in many languages (or even Web pages with many languages on the same page) if text on the Web pages is represented using glyphs rather than a text encoding standard.

Chapter 4

Methodology

A Web page has been created in which all of the Japanese text is represented by glyphs, which are defined to be image files containing images of text. Different image file formats have been tested, in order to find a file format which can be downloaded relatively quickly, but which also has sharp and legible text. This chapter describes the methods used to convert text into glyphs in different image formats, and to format the glyphs on Web pages for public download.

Original Materials

The Web pages created for this project are copies of Japanese language Web pages already available to the author. A total of twenty five pages were imaged, including almost all of the hierarchy of Web pages reachable below the Computer Pop for U page at http://www.kcg.ac.jp/35event/index.html[1] (this page is mirrored at http://www.kcg.edu/event1/index.html, and is also shown in Appendix B of this thesis). The Japanese text on most of these pages is coded using the ISO-2202-JP coding standard (one page is coded using shift-JIS), and all of the pages are impossible to read on any computer unless the computer has Japanese fonts and Japanese operating system extensions installed.

Imaging the Text

Web pages in the Computer Pop for U hierarchy were displayed on a freeBSD workstation,[2] using the Japanese language version of Netscape Communicator 4.0 for freeBSD. As a first step in image capture, the entire visible Web page was captured directly from the screen using the Grab feature of XV 3.10a.[3] The XV control panel and "grab" panel are shown in Appendix C. Pressing "grab" on the main control panel pops up the "grab" panel. After pressing "grab" on the lower panel, the user can grab a window with a left mouse click, or grab a rectangular region from anywhere on the screen using the middle mouse button of a three-button mouse.

Web pages too long to be viewed all at once were captured in several consecutive screen shots, with the consecutive screen shots arranged so that each paragraph of text was shown unbroken on at least one screen shot. Since captured images are harder to resize than text, most of the Web pages were captured after setting the Netscape Navigator window to a width of 610 pixels, so that users of a standard 640x480 VGA monitor will not have to scroll right and left to view the page.

The full-page, formatted screen shots were then cropped at the edges of each paragraph of Japanese text, in order to create two-color images containing nothing but text. The resulting glyph images were saved in both GIF and JPEG file formats. JPEG files were initially saved with quality ratings of 75 percent, 50 percent, and 25 percent, but blurring of the text was already visibly annoying at 50 percent, so all of the statistics quoted in this article use JPEG with a 75 percent quality rating. A total of 164 glyph files were created, as shown in Table 1.

Table 1 shows the total number of glyph files per Web page, the total size of the glyphs on a Web page, and the total download time over a 28.8 kbps modem. The "download time" of each image file is calculated by dividing the file size, in kilobytes, by the download speed in kilobytes of a 28.8 kbps modem, which is 28.8/8=3.6 kilobytes per second.

As shown in Table 1, the JPEG version of a glyph file, stored with a 75 percent quality rating, is three to five times larger than the equivalent GIF file. Even at a quality rating of 25 percent, with barely legible text, JPEG glyph files are larger than equivalent GIF files. This is as predicted by Greenspun,[4] who claims that the GIF format is efficient for storing images which are painted using a limited palette of colors, such as a glyph, while the JPG format is more efficient for smoothly varying images such as photographs.

HTML files (number of glyphs)	Size (K) GIF	Dowload time GIF (sec)	Size (K) JPEG	Dowload time JPEG (sec)
Index (13)	13.0	5	46.1	55
Cm/index (22)	17.4	10	105.1	66
Quiz/goods (12)	40.0	10	222.9	100
Quiz/index (6)	8.3	7	42.3	100
Station/index (12)	13.6	8	27.5	80
Student/index (6)	12.3	4	44.4	108
Student/jnp3D Index (5)	31.8	3	94.6	120
Student/tikitiki/ tikitikiTank11/index (2)	2.0	4	6.7	180
Student/tikitiki/ tikitikiTank2/index (24)	94.6	12	262.5	63
Student/tikitiki/ tikitikiTank3/index (1)	0.7	1	1.9	180
Topic/cs (9)	10.5	6	25.5	87
Topic/livenet (10)	17.9	8	127.5	87
Topic/staff (7)	9.2	7	20.0	129
Topic/wcpu (5)	13.8	8	103.2	132
Topic/wnew (7)	20.8	1	156.8	34
Topic/overseas (11)	8.5	6	41.6	60
Topic/boston (3)	4.3	6	19.5	200
Topic/china (3)	8.2	2	43.5	120
Topic/ganna (3)	6.0	1	29.3	80
Topic/ginba (3)	5.7	3	27.7	120
Topic/kenya (4)	11.5	1	61.8	80
Topic/peru (3)	5.4	3	25.7	120
Topic/po (3)	8.6	2	44.8	100
Topic/ro (5)	5.5	8	12.1	165
Topic/thai (3)	8.5	1	43.5	80
Average (6.8)	15.1	5.1	65.46	105.8
Standard Deviation (5.5)	18.7	3.3	66.09	41.9

Table 1: Glyph files per Web page

On the Web pages in this hierarchy, the total size of other graphics on the page (including photographs, icons, and computer animation) is often larger than the total size of the glyphs, so that converting the page from text to glyphs has a relatively small impact on download time. This is demonstrated in Table 2, which shows the total size of non-glyph graphic files included as either background or foreground image files on each of the converted pages.

The total size of a whole-page GIF image of each Web page, including both text and graphics, is given in Table 2. Notice that, in many cases, the whole-page GIF image file is considerably larger than the sum of all of the text and graphic image files on the page. It is reasonable that the whole-page image should be larger than the sum of all of the included image files, since the whole-page image must represent all of the included images, and all of the HTML layout formatting, using a single colormap and a single image file.

HTML files (number of glyphs)	Size of glyphs, GIF format (K)	Size of other graphics (K)	Whole-page GIF image (K)
Index (13)	14.1	41.7	91.2
Cm/index (22)	17.4	14.3	118.1
Quiz/goods (12)	40.0	46.2	281.1
Quiz/index (6)	8.3	23.5	101.2
Station/index (12)	13.6	93.7	542.6
Student/index (6)	12.3	8.0	44.0
Student/jnp3D Index (5)	31.8	6.4	81.3
Student/tikitiki/ tikitikiTank11/index (2)	2.0	6.4	55.0
Student/tikitiki/ tikitikiTank2/index (24)	94.6	26.1	310.8
Student/tikitiki/ tikitikiTank3/index (1)	0.7	6.4	32.1
Topic/cs (9)	10.5	70.5	234.2
Topic/livenet (10)	17.9	39.3	96.1
Topic/staff (7)	9.2	72.0	311.0
Topic/wcpu (5)	13.8	8.3	64.1
Topic/wnew (7)	20.8	9.7	61.0
Topic/overseas (11)	8.5	9.8	27.2
Topic/boston (3)	4.3	14.9	30.8
Topic/china (3)	8.2	23.8	41.7
Topic/ganna (3)	6.0	21.9	43.2
Topic/ginba (3)	5.7	31.3	40.6
Topic/kenya (4)	11.5	34.3	46.6
Topic/peru (3)	5.4	32.6	38.3
Topic/po (3)	8.6	25.1	43.3
Topic/ro (5)	5.5	44.3	62.9
Topic/thai (3)	8.5	13.5	39.5
Average (6.8)	15.1	29.8	113.5
Standard Deviation (5.5)	18.7	23.0	125.8

Table 2: Image and glyph file sizes

Web Page Formatting

The original text-based Web pages were converted to glyph-based Web pages by using Netscape Page Composer to replace each block of text with a pointer to the appropriate glyph image file. The program "wwwis" was used to fill in the correct WIDTH and HEIGHT tags for each image, so that the viewer's browser can lay out the Web page correctly before all of the images have been downloaded.[4] The resulting glyph-based Web pages were viewed using Netscape in order to confirm that the glyph-based and text-based pages have similar formats.

Most of the glyphs were images of single paragraphs or single lines of text, and none of the glyphs included more than one hyperlink. Hyperlinks were therefore tied to the entire glyph image, rather than just the key words; a user who clicks anywhere on the glyph image is taken to the address of the highlighted hyperlink text. Since the computer views a glyph as a regular image file, while the reader presumably views a glyph as a paragraph of text, hyperlink placement revealed some unusual formatting problems. Specifically, most browsers place a colored border around any image hyperlink, which is distracting to a person who thinks the image is a paragraph of text; this colored border was eliminated using the HTML tag BORDER=0.

Conversion from Text to Glyphs: Efficiency Considerations

Companies interested in publishing glyph-based Web pages will first want to know how long it takes to create a glyph-based Web page. For that reason, statistics are given in Table 3 for the creation of the twenty five example Web pages. Statistics are given only for the creation of GIF-based Web pages; the time to create a JPEG glyph-based page would be similar, but can be reduced somewhat if both GIF and JPEG

pages are created at the same time. All per-page statistics are rounded off to the nearest minute, and are therefore not necessarily very precise.

The second column of Table 3 shows the amount of time required to capture and crop all of the glyphs on a page and save them in GIF format. The third column shows the amount of time required to reformat the Web page to include glyphs in place of text. The fourth column shows the total glyph-page creation time for each page. The last column of the table shows the time required per glyph, in seconds, on each of the pages; this entry is calculated by dividing the total time for each page (column 4) by the number of glyphs on the page (column 1, in parentheses). The bottom two rows of the table show the average and standard deviation of each of the columns. The average of the last column is 90.8 seconds, but if we divide the average time per page (9 minutes) by the average number of glyphs per page,[5,6] we get an "average time per glyph" which is somewhat lower: approximately 79 seconds.

HTML files (number of glyphs)	Create glyphs (minutes)	Convert HTML (minutes)	Total min/page	Seconds per glyph
Index (13)	12	5	10	50
Cm/index (22)	12	10	21	63
Quiz/goods (12)	7	10	19	95
Quiz/index (6)	3	7	9	90
Station/index (12)	7	8	15	75
Student/index (6)	3	4	8	96
Student/jnp3D Index (5)	2	3	5	100
Student/tikitiki/ tikitikiTank11/index (2)	1	4	5	150
Student/tikitiki/ tikitikiTank2/index (24)	10	12	22	60
Student/tikitiki/ tikitikiTank3/index (1)	1	1	2	120
Topic/cs (9)	5	6	12	80
Topic/livenet (10)	4	8	12	80
Topic/staff (7)	7	7	14	120
Topic/wcpu (5)	2	8	10	120
Topic/wnew (7)	2	1	3	26
Topic/overseas (11)	4	6	10	55
Topic/boston (3)	3	6	9	180
Topic/china (3)	3	2	5	100
Topic/ganna (3)	2	1	3	60
Topic/ginba (3)	1	3	5	100
Topic/kenya (4)	2	1	3	60
Topic/peru (3)	2	3	5	100
Topic/po (3)	2	2	4	80
Topic/ro (5)	2	8	10	150
Topic/thai (3)	2	1	3	60
Average (6.8)	3.9	5.1	9.0	90.8
Standard Deviation (5.5)	2.9	3.3	5.8	35.5

Table 3: Time required to create each of the glyph-based Web page

Summary

The HTML document trees described above are available for public download. The original text-based Web pages are available at http://www.kcg.edu/event1/index.html. The GIF-formant glyph-based Web pages are available at http://www.kcg.edu/event2/index.html. The JPEG-format glyph-based Web pages are available at http://www.kcg.edu/event3/index.html .

Endnotes for Chapter 4

[1] Yamada, N., Nishikado, M., Tani, N., and Kusumoto, H. "Computer Pop for U." Kyoto, Japan: Kyoto Computer Gakuin, 1998. http://www.kcg.ac.jp/35event/index.html

[2] Wolfram Schneider, "FreeBSD Home Page." Berkeley, CA: FreeBSD Inc, 1999. http://www.freebsd.org/index.html.

[3] Bradley, John. XV Interactive Image Display for the X Windows System. Bryn Mawr, Pennsylvania: John Bradley, 1994

[4] Greenspun, Philip. Philip and Alex's Guide to Web Publishing. New York, NY: Morgan-Kaufman, 1999.

[5] Benett, Gordon. Introducing Internets. Hollis, New Hampshire: Que Corporation, 1996

[6] International Standards Organization, "Information processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1." ISO Standard 8859-1, 1987.

Chapter 5

Evaluation of the Glyph-Based Pages

In order to evaluate the ability of users to navigate and read the glyph-based Web pages, a user interaction experiment was conducted at Kyoto Computer Gakuin (Kyoto School of Computer Science, Kyoto, Japan). This section reports the methodology and results of the experiment.

Methods

Students, staff, and other volunteers were recruited at the Kyoto Computer Gakuin (KCG) to participate in an ease-of-use experiment. Some of the volunteers were relatively naive computer users, and some had many years of internet experience, but all volunteers had some experience with basic computer user interface. Volunteers were given the following list of questions, and asked to find the answers as quickly as possible by browsing through a specified HTML document tree. The questions given to volunteers were:

1. What was the name of the event which KCG sponsored on May 4 of this year at the Kyoto station Muromachi plaza?

2. What time did the event start? What time did it end?

3. Where was the "Cyberstation" event held? When was it held? Why was this particular day chosen?

4. KCG co-sponsored the event with a radio station. What was the name of the radio station?

5. Write the names of a few of the people who staffed the cyberstation event.

6. In connection with the May 4 event, KCG offered an on-line quiz. Name six of the prizes.

7. What band played at the Muromachi plaza event?

8. KCG has sent students and instructors to several foreign countries to support computer education in those countries. What are the countries?

9. Please name some of the KCG TV and radio commercials which you can listen to over the internet.

10. How many student-written programs are listed on the "KCG students" page," and what are the names of the programs?

Fifteen volunteers were given the original text-based tree to browse, and fifteen were given the GIF-format glyph-based tree to browse. Volunteers were scheduled in twenty-minute slots on a personal computer (PC) or Macintosh. The Netscape cache was cleared after every volunteer, so that each volunteer would be required to download all of the desired Web pages from the server.

Ease of Use: Quantitative Measures

Volunteers for the ease-of-use experiment were asked to answer the ten questions listed above as quickly as possible, using the provided HTML document tree. Fifteen volunteers were given the original text-based document tree, and fifteen were given a GIF-encoded glyph-based document tree. Their answers to the quiz were graded, with par-

tial credit given for partially correct answers. Table 4 lists the number of correct answers per question for students in each group. Group A answered the questions using a tree of Web pages coded using Japanese text; group B used a tree of Web pages coded using glyph image files in GIF format. There were 15 total responses in each group.

Question Number	Grop A (Text-Based Web Pages)	Grop B (Glyph-Based Web Page)
1	15	15
2	15	12.9
3	15	15
4	15	15
5	15	12.5
6	12.5	15
7	15	15
8	11.9	14.7
9	15	15
10	15	15
Avergage	14.4	14.5
Standard Deviation	1.2	1.0

Table 4 : Correct Answers per quiz question.

Ease of Use: Survey Results

After answering the speed quiz, participants in the ease-of-use experiment were asked to either agree or disagree with the statements listed in Table 5. Volunteers marked a seven-point scale between 1 (agree strongly) and 7 (disagree strongly.) Fifteen volunteers responded to the text-based HTML documents, and fifteen volunteers responded to the glyph-based HTML documents.

Servey Question	Text-Based Averages	HTML users Std. Dev.	Glyph-Based Average	HTML users Std. Dev.
1. These web pages take longer to download than normal web pages.	5.8	1.2	4.7	1.5
2. I found the layout of text and graphics to be confusing.	5.0	1.1	5.1	0.9
3. It was easy to find the hyperlinks.	2.8	1.8	4.4	2.7
4. Text sometimes shows up as strange meaningless characters, instead of legible Japanese.	4.2	2.6	5.6	2.2
5. The Japanese text is sometimes fuzzy and hard to read.	5.5	1.5	4.7	1.4
6. Some of the pictures were not found, and the browser displayed an icon instead of the missing picture.	5.3	1.4	5.0	2.2
7. It was easy to find the answrers to the speed quiz questions.	4.3	1.5	3.1	2.2
8. I thought these web pages were easier to use than normal web pages.	3.0	0.9	2.7	1.0

Table 5: Opinion survey of Web page users

There were small differences between the experience reported by group A and the experience reported by group B, but it should be noted that the variability within each group was much larger than the small differences reported between groups. Group A was more likely to find strange characters in the text (question 4), and group A was able to find the hyperlinks more easily (question 3). Group B had slower download times (question 1), and the text was more likely to be fuzzy and hard to read (question 5), but group B found the Web pages to be slightly easier to use and better-designed than did group A (question 7 and 8).

Volunteer Comments

After answering the survey questions above, volunteers were allowed to view both kinds of Web pages (the "Group A" pages, which used Japanese text encoding to represent text, and the "Group B" pages, which used glyph images to represent text), and were asked to comment on the differences. Their comments are listed below. The difference between "Group A" pages and "Group B" pages was not described to volunteers, so some of their experiences may reflect variability in network congestion, or other kinds of variability unrelated to the actual Group A/Group B differences. For example, one of the subjects claims that the pictures took longer to load for Group B pages, but in fact the pictures on the Group B pages were identical to the pictures on the Group A pages.

Group A Comments

1. Cyberstation screen came up quickly. The icon beforehand took a bit long.

2. The first thing I noticed was slow access (A is faster). But I couldn't see the characters on the overseas page. Once I had seen a page, it was easy to bring it up again. The "From Overseas" page had altered characters, which was annoying. It should be fixed.

3. B feels a little slower.

4. This is a comparatively easy-to-use home page, I think.

Group B Comments

1. B feels a little slower than A, but I think it's still totally OK.

2. A was a little faster, but I had no trouble with B. Faster is better.

3. A was three times faster than B (especially the pictures).

Chapter 6

Summary and Conclusion

This thesis project describes a method for including Japanese text on Web pages so that it can be seen on computers which do not have Japanese fonts or Japanese display software installed.

Uses and Usefulness of the Methods in this Thesis

It might be argued that the advent of unicode will make the work in this thesis obsolete; when all of the computers in the world understand unicode, one might think that all computers will be able to display any human language without extra effort. Unfortunately, this is not the case. First, even when all new computers understand unicode, there will still be many older computers in use which do not understand unicode. Second, just because software understands unicode does not mean that it can display all of the unicode characters on screen. In order to display all of the characters in the unicode character set, a computer must have fonts available for all of those characters, which would require a huge commitment of disk space. Preliminary reports suggest that most unicode fonts will only be partial fonts, covering a limited subset of the possible unicode characters.[1] It is quite possible that a user in the United States might log on to a Japanese Web site using a program which is perfectly able to interpret unicode or JIS-based characters, but which is unable to display the Japanese text, because the computer does not have any Japanese fonts installed.

With the method proposed in this thesis, a Web site designer can create a Japanese-language Web site which can be viewed by any user, anywhere in the world, regardless of the fonts that user may have installed on his computer. The method is simple: the Web site designer converts each block of uninterrupted text into an image (a "glyph"), and positions the image on the page in place of the text. Statistics are presented to show that the conversion of an existing Web page into a glyph-based representation only requires about nine minutes per page: about four minutes to image the text, and about five minutes to reformat the HTML. Using an efficient GIF representation, the average size of the glyph images (15K) was half of the average size of all other images on the page (30K). Users of the glyph-based Web pages were slightly more likely to notice long download times, but they were also slightly more likely to enjoy reading the Web page; in both cases, differences between the two groups were much smaller than the differences between individual users.

Future Work

The method proposed in this thesis allows a Web designer to display text on a computer which does not have the required fonts by essentially bypassing the HTML standard: instead of coding text using standard HTML, the Web designer codes text in an image, which is downloaded separately for each Web page. This gives a Web designer better control over the appearance of his Web pages, but at the cost of portability: the glyphs you create for one Web page don't help you to make the other Web pages on your Web site.

A natural extension of the work in this thesis would be a system which allows each Web designer to publish the font file which he would like viewers to use when they

look at his Web pages. In this system, text on Web pages would be included as text, but with a specified font. In the current HTML standard, display font can be controlled using the "font" tag, for example

This is a test in the MyFont font.

The problem with the existing "font" tag is that, if the user's computer does not have the font "MyType,MyFont" installed, it tries to display the text in some default font. If the text is not in English, the display will often fail, because the default font is usually an English font. A better solution would be to allow Web publishers to specify a location from which fonts can be downloaded, perhaps using an href option in the font tag:

This is a test in the MyFont font.

The Web designer would then publish the MyFont.tt file in the specified directory. When a browser program downloads his Web page, the browser program would notice that it does not have the MyFont.tt file, so it would download the font, and store it in the browser cache. Then, from that time on, until he clears his browser cache, the user would be able to download and view any Web page written in the MyFont font just as quickly as if the page was written in English.

From the point of view of the Web designer, this method would solve all of the same problems solved in this thesis, but with much more portability. With this method, the

Web designer can define any unusual characters he needs in his documents, and include them in the font file. The characters do not need to be characters in a common language, or even in any existing language; the Web designer could even make up a language with its own writing system, if he wants to. Once he has created the font file, all of the other Web pages on his Web server can be written, stored, and edited if necessary as if they were simple English text (coded in ASCII if the number of characters is small, or in unicode if the number of characters is large).

Endnotes for Chapter 6

[1] Penney, Laurence, "TrueType and Unicode," TrueType Typography. Type*chimerique, 1996. http://www.truetype.demon.co.uk/unicode.htm

Bibliography

Abramson, Dean, "Globalization of Windows," Byte Magazine. New York, NY: McGraw-Hill, November 1994. http://www.byte.com/art/9411/sec9/art7.htm

Benett, Gordon. Introducing Internets. Hollis, New Hampshire: Que Corporation, 1996

Boutell, Thomas ed., "PNG (Portable Network Graphics) Specification." Cambridge, MA: World Wide Web Consortium, October, 1996. http://www.boutell.com/boutell/png/index.html

Bradley, John. XV Interactive Image Display for the X Windows System. Bryn Mawr, Pennsylvania: John Bradley, 1994.

Ellsworth, Jill H. , and Ellsworth, Matthew V. Internet Business Book, New York, NY: John Wiley and Sons, Inc., 1996.

Fowles, Ken. "Developing TrueType." Redmond, Washington: Microsoft Corporation, June 1997. http://www.microsoft.com/truetype/unicode/cs.htm

Greenspun, Philip. Philip and Alex's Guide to Web Publishing. New York, NY: Mogan-Kaufman, 1999.

Harley Hahn. The Internet Complete Reference. Berkeley, California: Osborne McGraw-Hill, 1996.

Hars, Adele. "Organizing Babylon," Byte Magazine. New York, NY: McGraw-Hill, March 1996. http://www.byte.com/art/9603/sec18/art1.htm

International Standards Organization, "Information processing -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1." ISO Standard 8859-1, 1987.

International Standards Organization, "Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane." ISO Standard 10646-1, 1993

Japanese Standards Association, "Code of the Japanese graphic character set for information interchange," JIS X 0208-1978, -1983 and -1990.

Kalakota, Ravi. and Whinston, Andrew B. Electronic Commerce. Essex, England: Addison Wesley Longmon, 1997.

Lane, Thomas, ed. "JPEG image compression FAQ." Independent JPEG Group, October, 1997. http://www.cis.ohio-state.edu/hypertext/faq/usenet/jpeg-faq/index.html

Miller, L. Chris, "Transborder Tips and Traps," Byte Magazine. New York, NY: McGraw-Hill, June 1994. http://www.byte.com/art/9406/sec7/art2.htm

Murai, Jun, Crispin, Mark, and van der Poel, Erik M., "Japanese Character Encoding for Internet Messages," RFC 1468. Los Angeles: RFC Editor, Information Sciences Institute, June 1993. http://www.faqs.org/rfcs/rfc1468.html.

Penney, Laurence. "TrueType and Unicode," TrueType Typography. Type*chimerique, 1996. http://www.truetype.demon.co.uk/unicode.htm

Poskanzer, Jeff. "ppmtopgm manual page." December, 1988.

Randall, Nail. The Soul of the Internet. London, England: International Thomson Computer Press, 1997.

Turnbull, Stephen. "Alphabet Soup: The Internationalization of Linux, Part 1." Linux Journal. Seattle, WA: Specialized Systems Consultants, Inc., March 1999.

Unicode Consortium. Unicode. San Jose, California: Unicode, Inc., 1992-1998. http://www.unicode.org/index.html

Wolfram Schneider, "FreeBSD Home Page." Berkeley, CA: FreeBSD Inc, 1999. http://www.freebsd.org/index.html.

Washburn, Kevin. and Evans, Jim. TCP/IP: Running a Successful Network. Essex, England: Addison Wesley Longmon, 1996.

Yamada, N., Nishikado, M., Tani, N., and Kusumoto, H. "Computer Pop for U." Kyoto, Japan: Kyoto Computer Gakuin, 1998. http://www.kcg.ac.jp/35event/index.html

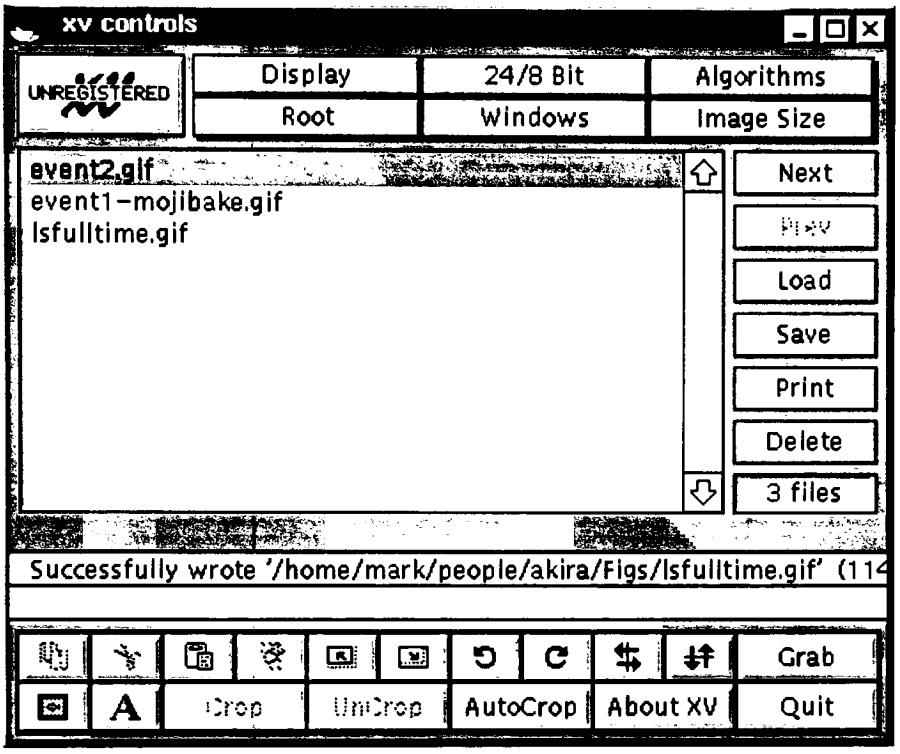
Appendix A

The KCG special event Web page, as viewd on a Lation-1 computer.

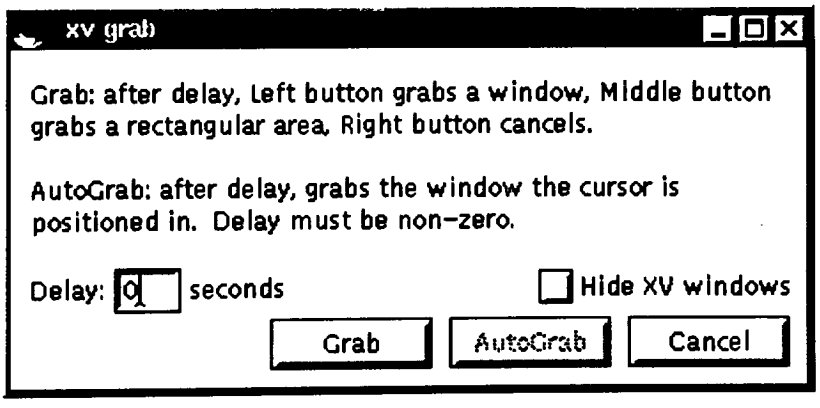
Appendix B

Glyph-based version of the KCG special event Web page.

Appendix C

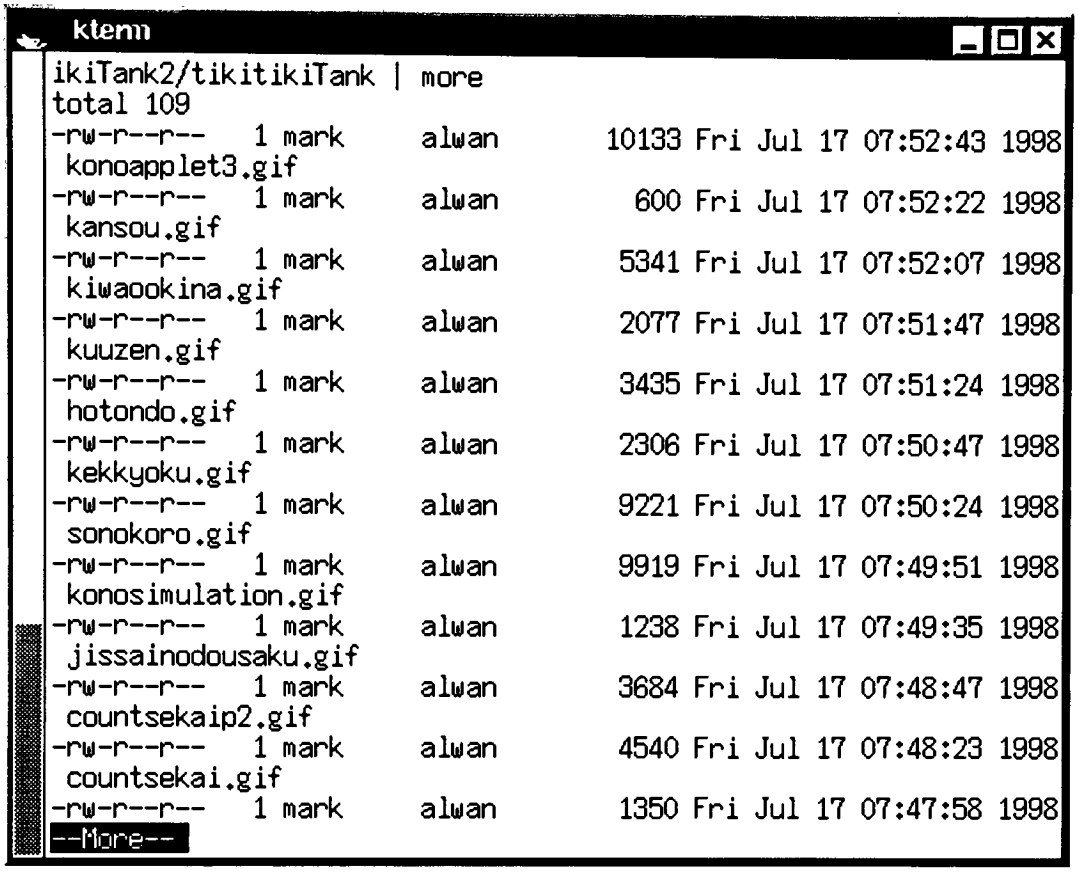


The VX control panel.



The VX grab control panel.

Appendix D



Listing from the Unix Filesystem, showing creation times of 12 glyph files.